



UI/UX Considerations

When there is no keyboard or mouse

By Quentin “Q” Alldredge
Q Software Innovations



About Q

Feel Free to call me “Q”

- I got the nickname for both the Star Trek and James Bond references
- Work of the Aerospace and Defense Industry
 - First at ATK (Now Northrup Grumman)
 - Now at Hill Air Force Base
- LabVIEW Consulting as Q Software Innovations

Contact Q

Phone/Text: +1 (435)-730-1198

Email: q@qsoftwareinnovations.com

Website: www.qsoftwareinnovations.com

LinkedIn: www.linkedin.com/in/quentin-q-allredge

LabVIEWWiki.org: Q

Twitter: [@QSI_Q](https://twitter.com/QSI_Q)

NI Community Forums: TheQ

LAVA Forums: The Q

stackoverflow: TheQ





Overview

- Requirements
- Types of Touchscreens
- Definitions
- UI Considerations
- UX Considerations
- Summary

Requirements

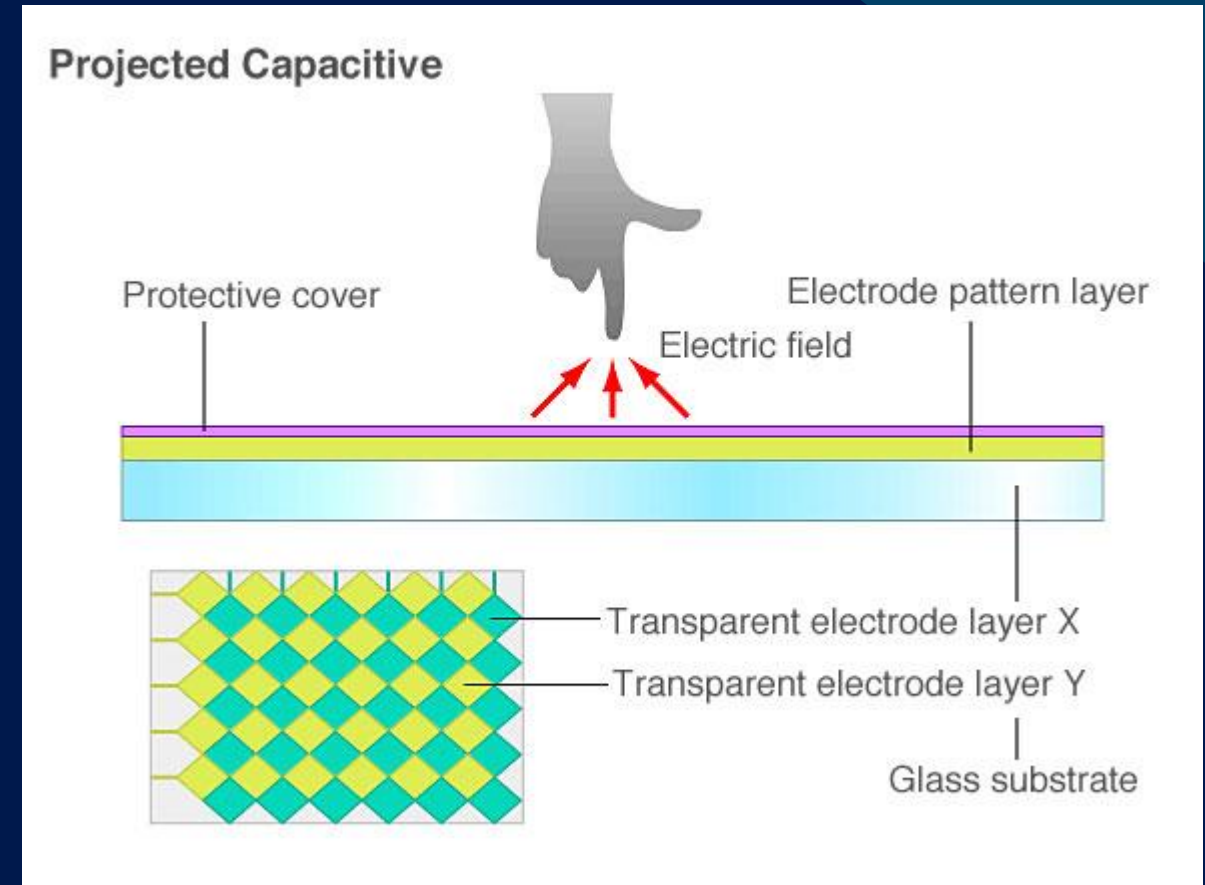
- Upgrade system from Laptop style to Touchscreen Tablet style
- Comply with MIL-STD-1472 Human Engineering
 - Alphanumeric Keyboard Minimum Size 0.5-inch x 0.5-inch
 - Other UI Component Minimum Size 0.65-inch x 0.65-inch
 - UI Component Minimum Spacing 0.25-inch
- Environmental Conditions
 - User must be able to use where gloves



Types of Touchscreens

Capacitive Screens

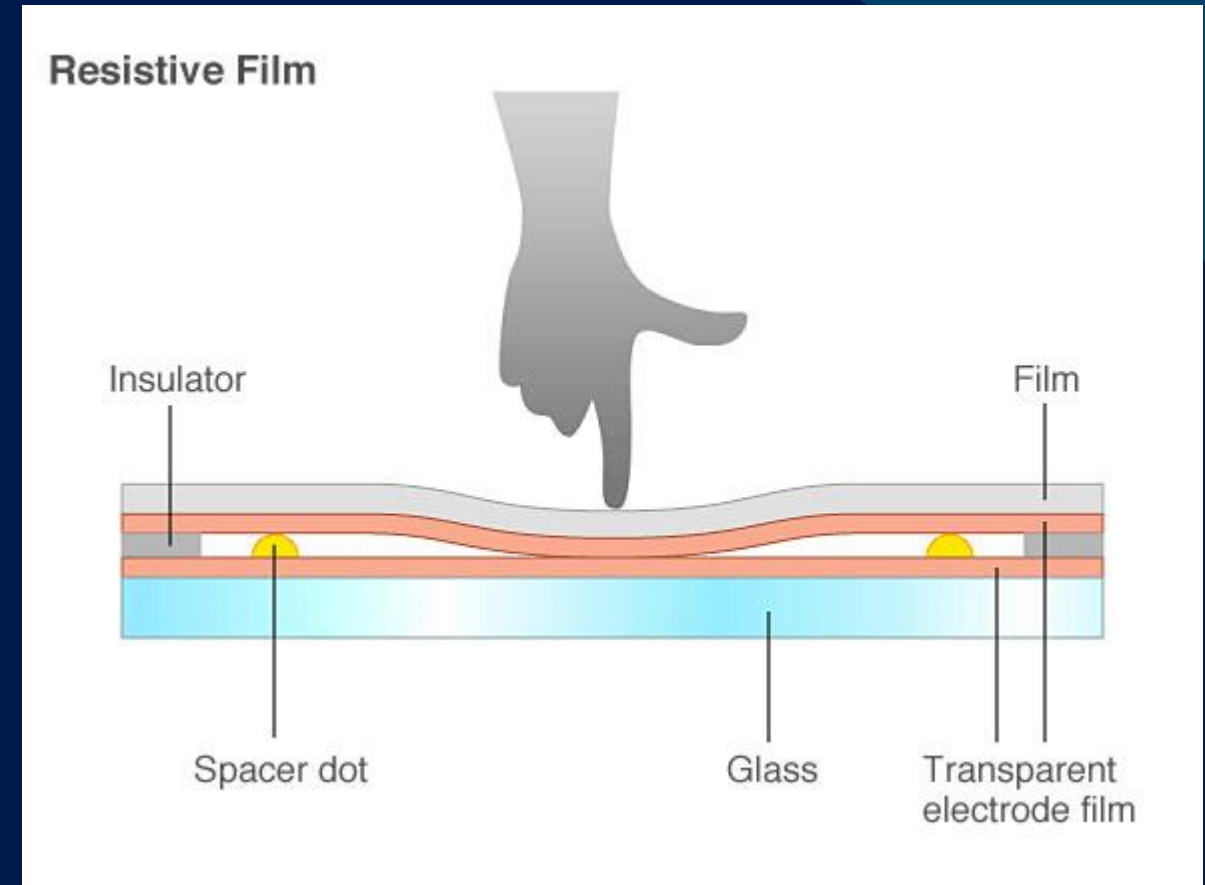
- Pros
 - No force required
 - Accurate
 - Good display quality
- Cons
 - Touch sensing doesn't work with gloves
- Examples
 - Modern smartphones and tablets



Types of Touchscreens

Resistive Screens

- Pros
 - Can use with gloves/stylus
 - Accurate?
- Cons
 - Requires force to sense input
 - Lower display quality due to two film layers
 - No multi-touch
- Examples
 - Older PDA (i.e. Palm Pilot)
 - Industrial Displays





Definitions

User Interface (UI)

The UI is the point of human-computer interaction and communication in a device. This can include monitors, keyboards, a mouse and the **appearance of the application**.

User Experience (UX)

The UX is the overall experience of a person using a software product such as a website or computer application, especially in terms of how **easy or pleasing it is to use**.

These are not interchangeable terms, but they are also not mutually exclusive either.



Definitions

User Interface (UI)

Make User/Software Interactions **Accessible**.

User Experience (UX)

Make User/Software Interactions **Easier**.



UI Considerations

Making User/Software Interactions
Accessible



UI Considerations

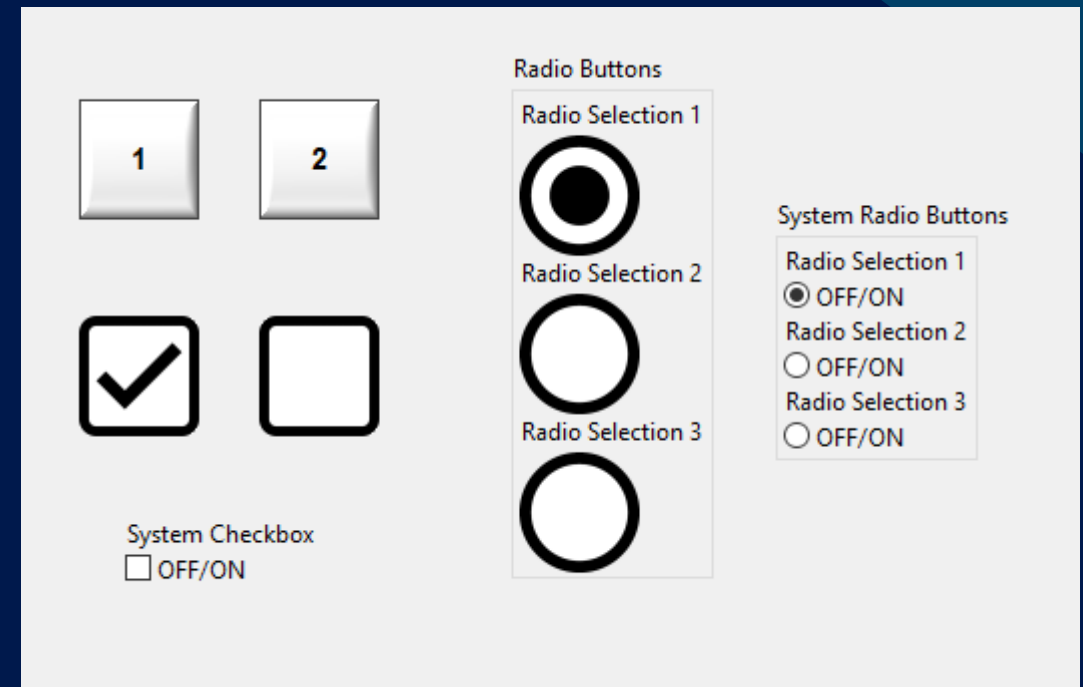
Controls

- Most regular Controls are too small for this application
 - Checkboxes
 - Radio Buttons
 - Lists
 - Scrollbars
 - Menus
 - Dialogs
 - On-Screen Keyboard

UI Considerations

Buttons, Checkboxes, and Radio Buttons

- **Size:** Minimum 0.5-inch x 0.5-inch
 - 1024x768 at 10.8-inch screen (9.41-inch x 5.29-inch) → 119 pixels/inch
 - Minimum \approx 60 pixels x 60 pixels
- **Spacing:** Minimum 0.25-inch
 - Minimum \approx 30 pixels
- **Made library of custom controls with Strictly Typed Buttons**



Front Panel grid spacing at 12 pixels



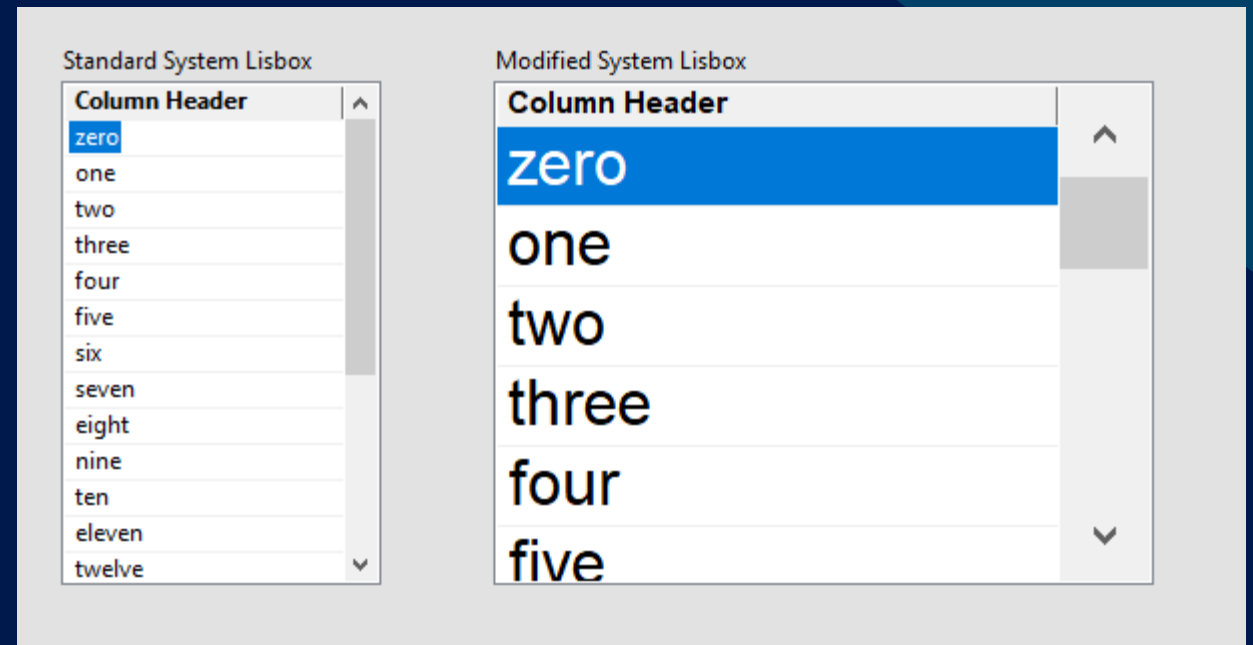
UI Considerations

Listboxes for Selection

- **Standardized on:**

- **Font:** Arial
- **Column Header:** 18pt Bolded
- **Body Text:** 36pt font for selectable text if single row
- **Color:** Black regular, White in selection
- **Highlight Entire Row:** True
- **Control Style:** System Controls

- Create **Scripts (VI Scripting)** to set Standard and/or **VI Analyzer Tests** to check Standard is followed.





UI Considerations

Scrollbars: Listboxes, Trees, and Tables

- **Customize the Control**
 1. Create a new Control (.ctl) file
 2. Add the control {Listbox, Multicolumn Listbox, Tree, or Table}
 3. Change to Customize Mode
 4. Select and Drag Parts to Size {Table/Tree Area, Scrollbar(s), Corner Box}
- **Alternatively:** You can change the Scrollbar Setting, pre-Windows 10 in:
 1. Control Panel -> Appearance and Personalize -> Change window glass colors -> Advanced appearance settings.
 2. Select Item = Scrollbar, change the size.



UI Considerations

Scrollbars: The INI Flag

- If you do change the scrollbars this way,
DON'T FORGET TO ADD THIS TO THE LabVIEW.ini FILE

autoResizeScrollbars=FALSE

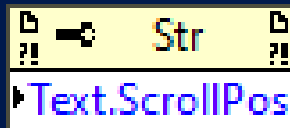
- If you don't, next time you open all scrollbars will reset to default size
- **You don't know pain until you must go through 50 modules fixing this...**
- Fortunately, we were able to script a search and replace for all these controls



UI Considerations

Scrollbars: Strings and other uses

- **Not customizable** like Listboxes, Trees, and Tables
- First used **XControls...** then **crashiness**
- Invented **QControls...** then **awesomeness**
- Used **LargeScrollbar QControl** to control String scroll position via property **Text.ScrollPos**



Note: You will need the QControl Toolkit to run this.

String

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Et malesuada fames ac turpis egestas maecenas pharetra convallis. Semper auctor neque vitae tempus quam pellentesque nec nam aliquam. Nisl nunc mi ipsum faucibus. Viverra tellus in hac habitasse platea dictumst. Ante metus dictum at tempor commodo. Dignissim suspendisse in est ante. Dolor sit amet consectetur adipiscing elit ut aliquam purus sit. Ipsum dolor sit amet consectetur. Odio eu feugiat pretium nibh ipsum consequat nisl vel pretium. Sociis natoque penatibus et magnis dis parturient montes.

Tristique risus nec feugiat in fermentum. Dictum varius dui at consectetur. Dui nunc mattis enim ut tellus elementum. Molestie ac feugiat sed lectus vestibulum mattis ullamcorper velit. Proin libero nunc consequat interdum varius sit amet. Feugiat in fermentum posuere urna nec tincidunt. Eget velit aliquet sagittis id consectetur purus ut faucibus pulvinar. Commodo sed egestas egestas fringilla phasellus faucibus scelerisque. Nulla malesuada pellentesque elit eget gravida. Quis vel eros donec ac odio tempor. Luctus venenatis lectus magna fringilla urna porttitor rhoncus dolor. Morbi quis commodo odio aenean sed adipiscing diam donec adipiscing. Tincidunt dui ut ornare lectus sit.

Lectus sit amet est placerat in egestas. Magna fermentum iaculis eu non diam. Viverra aliquet eget sit amet tellus cras. Fermentum leo vel orci porta non. Sed sed risus pretium quam vulputate. Penatibus et magnis dis parturient montes nascetur ridiculus mus mauris. Elit ut aliquam purus sit amet. At erat pellentesque adipiscing commodo elit at imperdiet dui accumsan. In fermentum posuere urna nec tincidunt. Convallis aenean et tortor at risus viverra adipiscing. Ut ornare lectus sit amet est placerat in egestas erat. Pharetra diam sit amet nisl suscipit. Enim sed faucibus turpis in eu mi bibendum neque. Morbi tincidunt ornare massa eget egestas purus viverra. Tincidunt praesent semper feugiat nibh sed pulvinar proin gravida. Proin libero nunc consequat interdum varius sit. Semper feugiat nibh sed pulvinar proin gravida hendrerit lectus a. In ornare quam viverra orci. Quam lacus suspendisse faucibus interdum posuere lorem ipsum. Amet commodo nulla facilisi nullam.

Est sit amet facilisis magna etiam tempor orci eu. Est ullamcorper eget nulla facilisi etiam dignissim diam. Tempor commodo ullamcorper a lacus vestibulum sed arcu non. Nisi lacus sed viverra tellus in hac habitasse platea dictumst. Nibh cras pulvinar mattis nunc sed blandit. Enim facilisis gravida neque convallis. Elementum pulvinar etiam non quam lacus suspendisse faucibus interdum. Volutpat lacus laoreet non curabitur gravida arcu ac tortor dignissim. Suspendisse ultrices gravida dictum fusce ut placerat orci. Purus sit amet luctus venenatis lectus magna fringilla. Ultricies integer quis auctor elit sed vulputate mi sit. Id neque aliquam vestibulum morbi blandit cursus risus at ultrices. Facilisi morbi tempus iaculis urna id volutpat lacus laoreet. Pretium fusce id velit ut. Ipsum nunc aliquet bibendum enim facilisis gravida neque convallis.

Ennetar dui id ornare arcu. Nunc non blandit massa enim nec dui nunc. In hac habitasse platea dictumst vestibulum rhoncus.

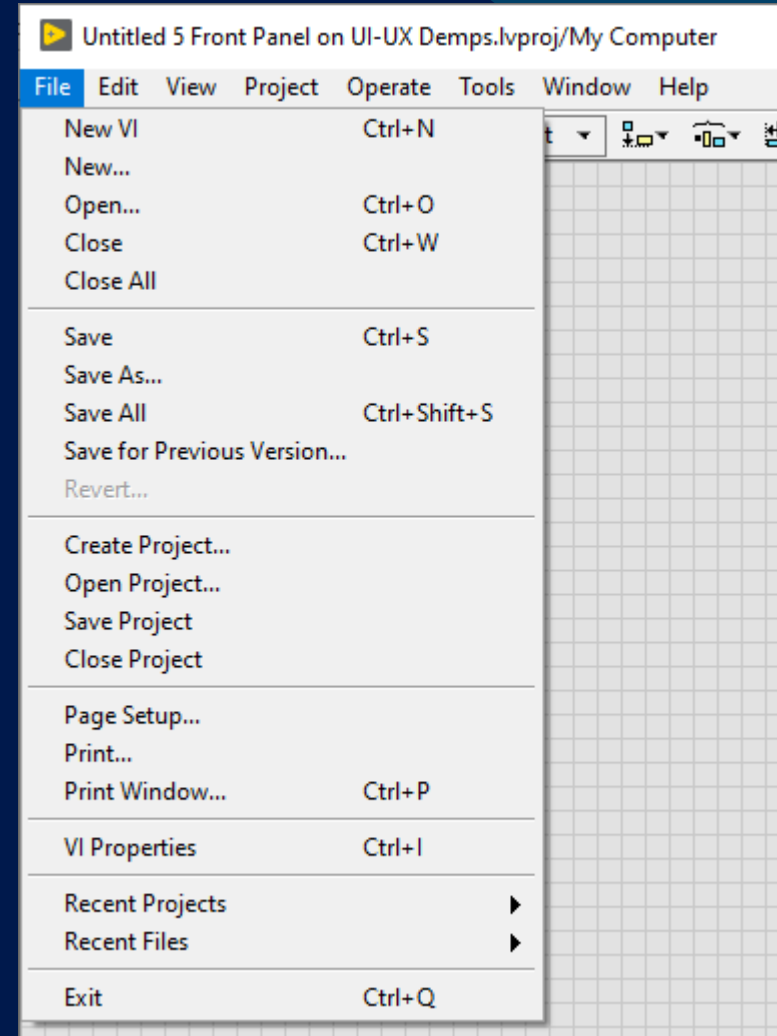
Toggle String's Built-in Scrollbar off



UI Considerations

Menus

- Menu text is too small
- Difficult to select near edges

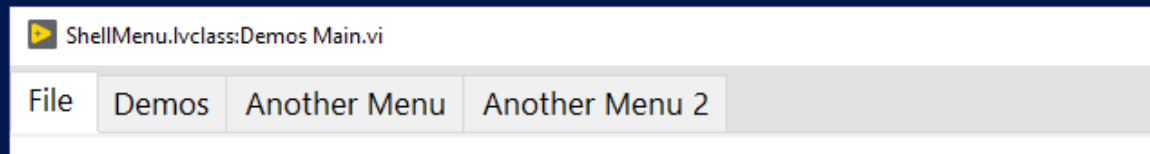




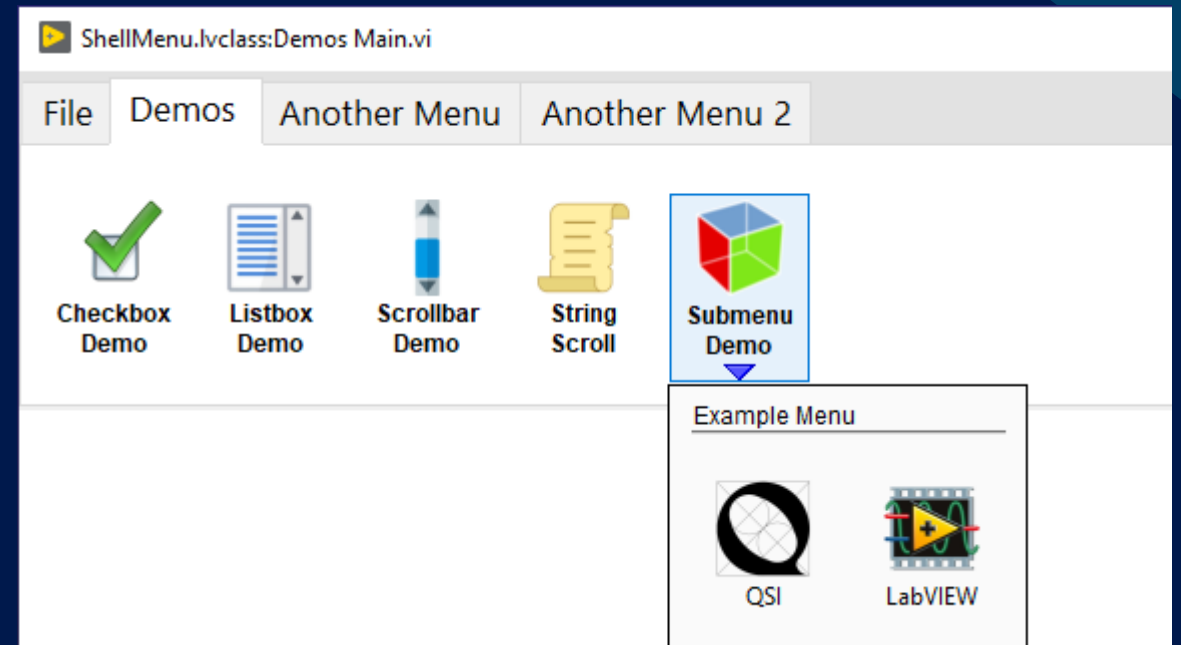
UI Considerations

Menus

- Used Tab Control, Buttons, Splitter, and popup VIs



Menu as Tab Control cutoff by Splitter Bar



Menu Open with Submenu Popup

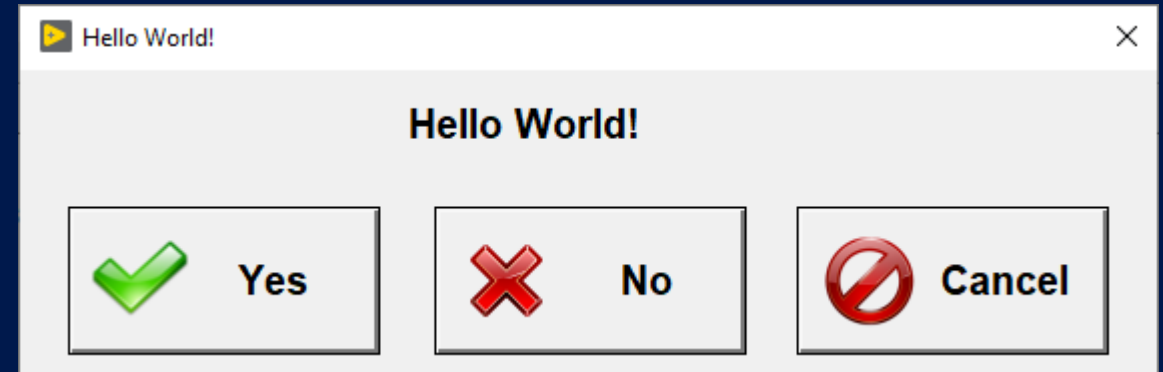
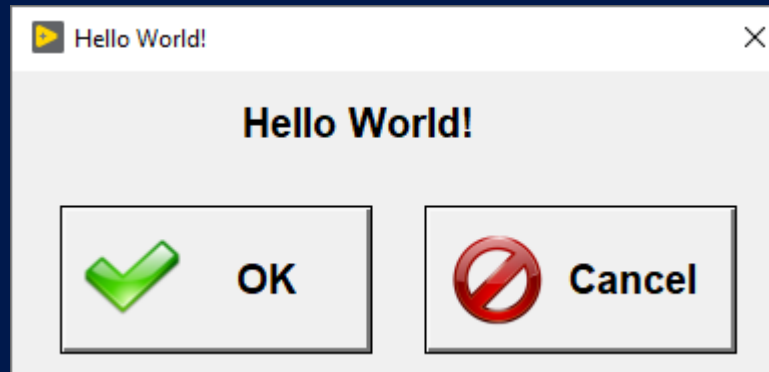
UI Considerations

Dialogs

- Standard Dialogs have too small of buttons



- Created Dialogs reuse library

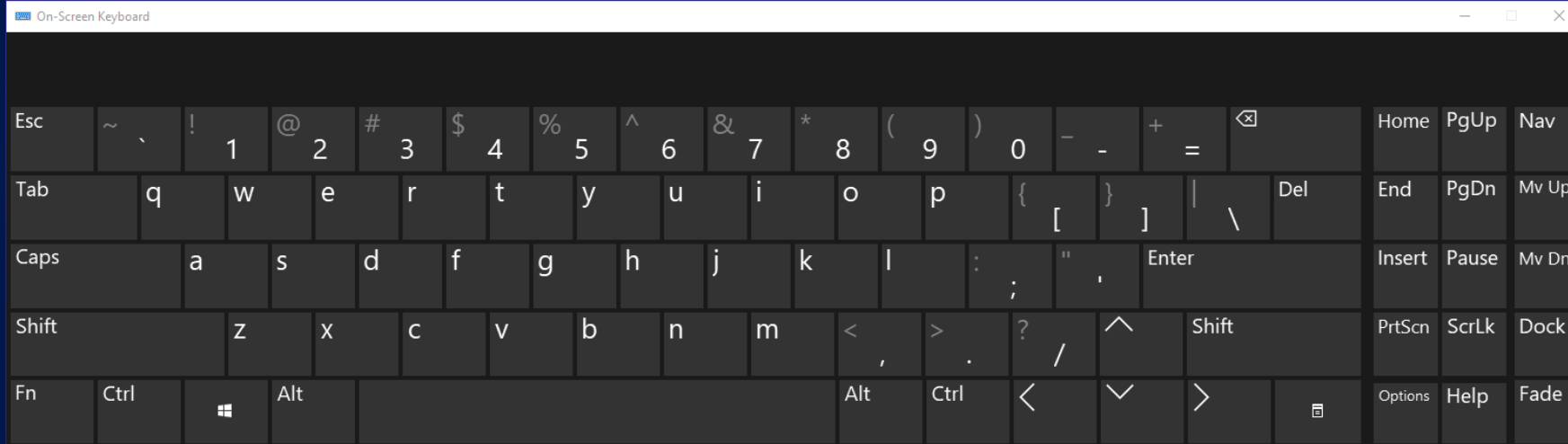




UI Considerations

Keyboard

- Built-in OS Keyboard is too small
 - Keys too small
 - Does not fulfill spacing requirements between keys





Enter Example String:

1234567890-=[X

qwertyuiop[]\

↑asdfghjkl;'

↑zxcvbnm,./↵

Something

Select

Select All

Paste

12

-=[X

Something

Cut

Copy

Paste

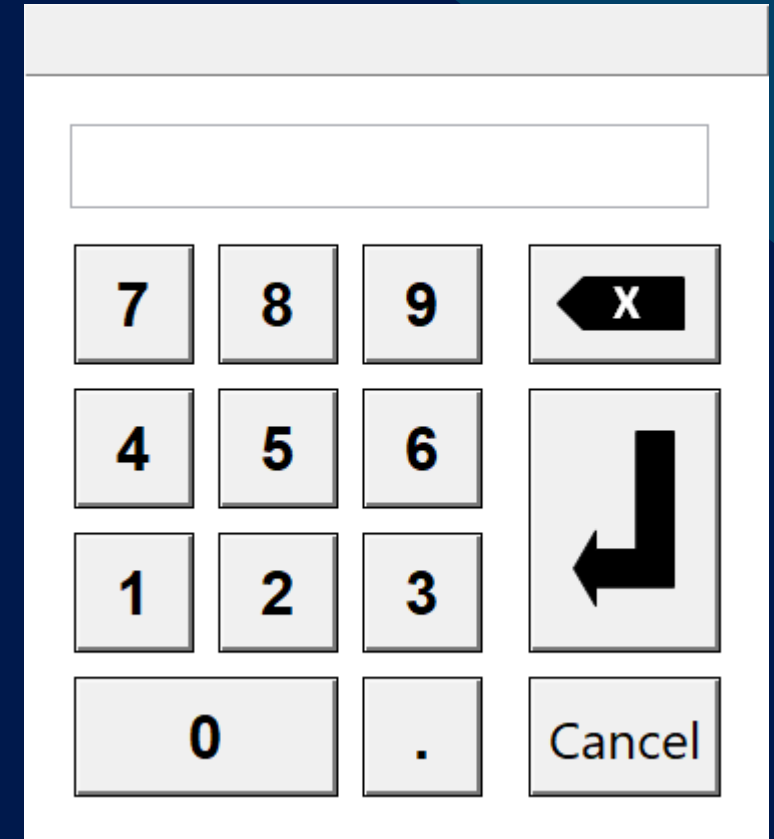
12

-=[X

UI Considerations

Keyboard: Lessons Learned

- Implemented as popup dialogs
 - Does not implement actual key presses through the OS
 - Edits the values of the controls directly
- Ended up having 3 keyboard types:
 - Standard Alphanumeric
 - Standard launches to allow field to be visible without being off the screen, if possible, but was movable
 - Must be launched asynchronously to show the entry as its typed
 - Full Screen Alphanumeric
 - Cover the screen
 - Provides a visible field as part of the keyboard
 - Numeric Only
 - Work essentially the same as the Standard





UX Considerations

Making User/Software Interactions **Easier**



UX Considerations

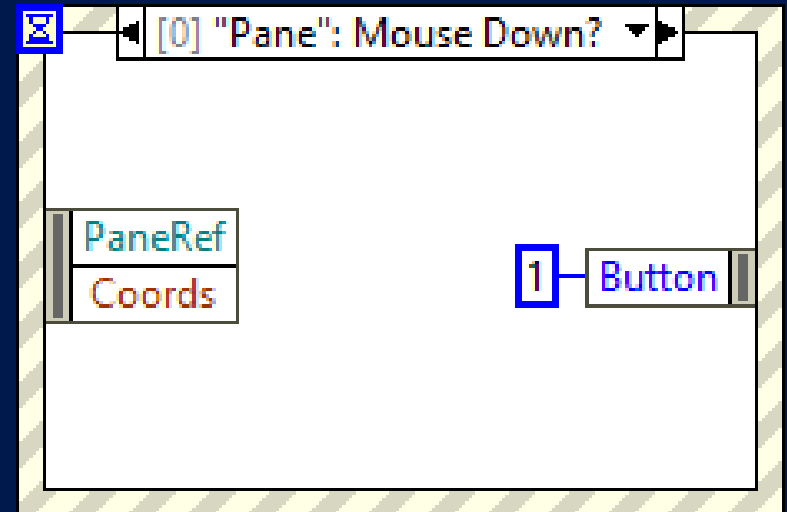
No Keyboard Navigation

- Keyboard navigation doesn't make sense or doesn't work well
 - No need Enter/Return for Buttons on Dialogs or otherwise
 - No need for Tab order to Tab through controls
 - No need for Alt+Keys for Menus/Buttons
 - No numbers for Selection from lists
- Work only with the system On-Screen Keyboard
 - Won't work with custom keyboard unless the functionality is purposely added
- Therefore, this is not for the sight impaired

UX Considerations

Right Clicks

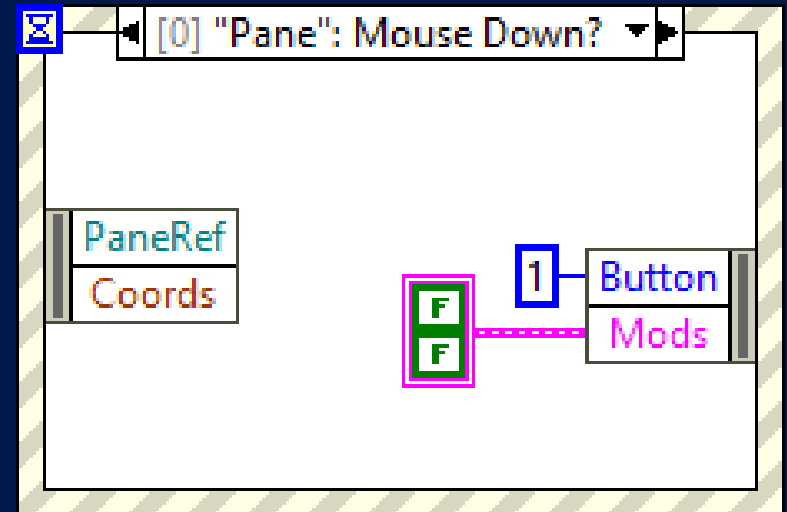
- Right clicks on touchscreens is just a **touch and hold a little longer**
- Its **confusing** when right click is triggered accidentally
- Instead **do not have any right-click menus**, make everything available through other means
- To **safeguard against accidental right clicks**, use the Mouse Down? filter event
 - Converts Right clicks to Left clicks
 - **Executes before Mouse Down notify event** which would still execute but now as a Left click
 - **Pane Mouse Down** events occur before control Mouse Down Events



UX Considerations

Double Clicks

- Same as right clicks:
 - Difficult to execute on purpose
 - Easy to accidentally execute or execute single clicks or right clicks instead
- To **safeguard against accidental double clicks**, add to the Mouse Down? filter event





UX Considerations

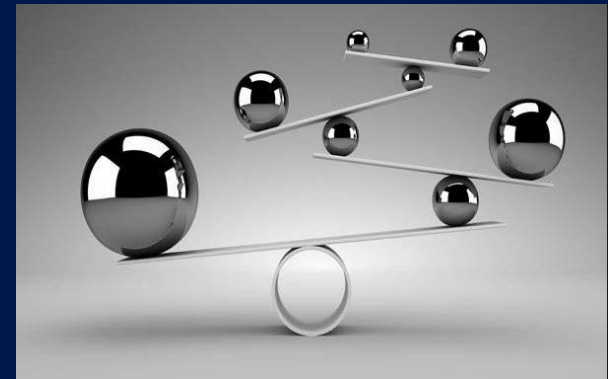
Minimize Text Entry

- **Selectable Lists** – Large list won't fit on screen, requires scrolling
- **Enums/Rings** – Smaller list, should fit on screen, shouldn't require scrolling
- **Combo Boxes** – Good if not all options are covered, then require text entry
- **Numeric Controls** – Use Increment/Decrement Buttons, consider programming large jumps if held down
- **Also Numeric Controls** – Use sliders/knob if they fit the data type. Consider them the enums and ring of numerics.

UX Considerations

Minimize Clicks

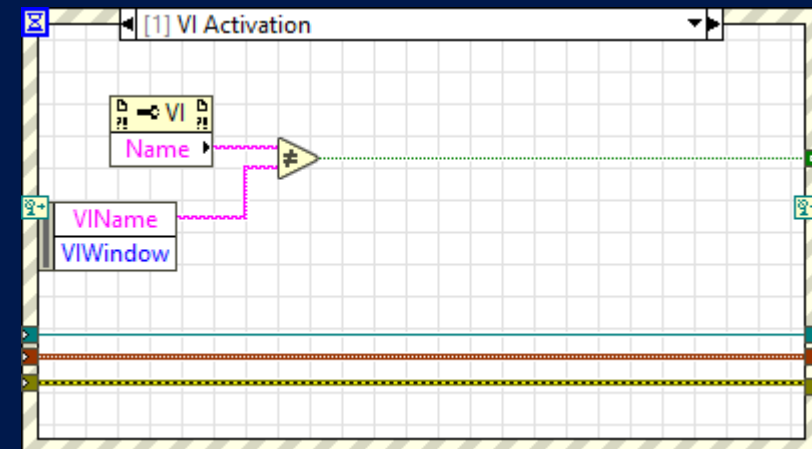
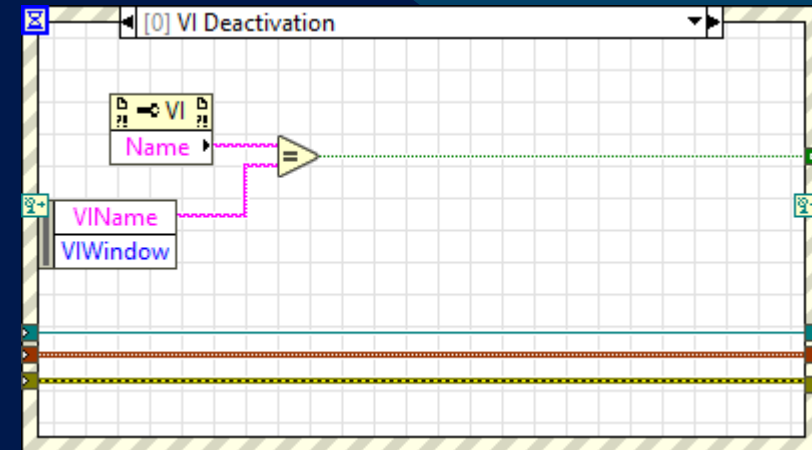
- Hard to define in bullet points
 - Think about how many clicks it takes to get to/performance functionality
 - Can this be reduced by button placement?
 - Can this be reduced by using a different control?
 - Can the multiple actions be merged?
 - Should multiple actions be separated to prevent mistakes/redo?
- Balance between **Ease of Use** and **Screen Real Estate**



UX Considerations

Dismissing Popups

- Usually done like the dialogs by OK/Cancel/"X" Buttons
- Can be done by simply clicking off the VI
- Don't have to dedicate space to buttons
- Accomplished through VI Activation or VI Deactivation Events
 - These are enabled through scripting but are available in the Run-Time Engine (can be used in an executable)

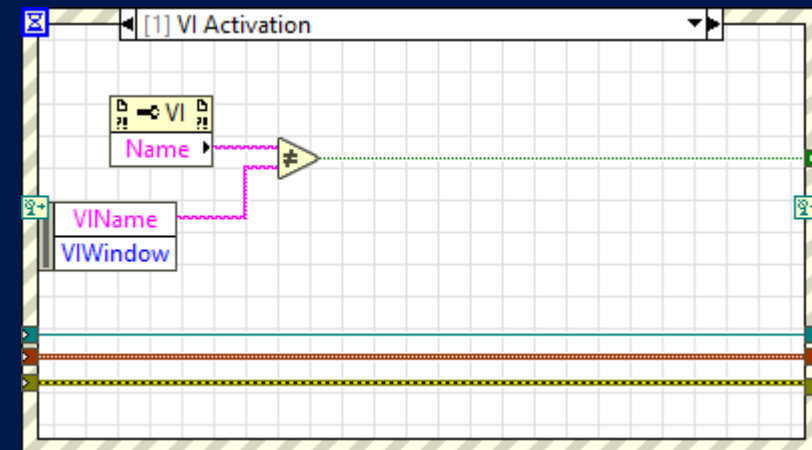
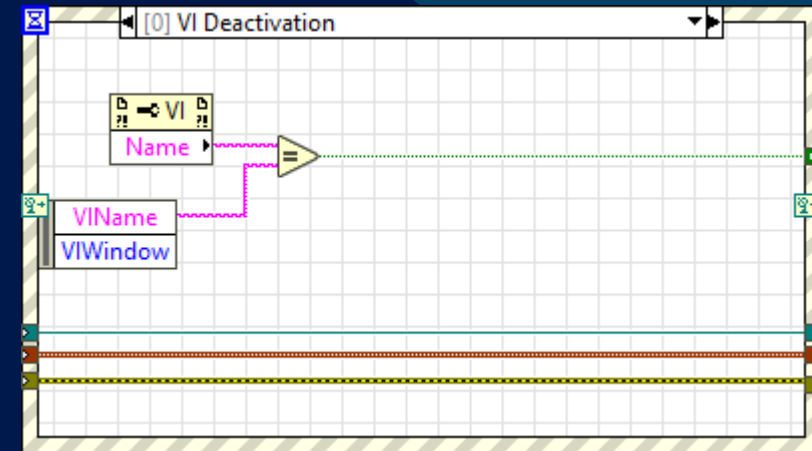


UX Considerations

Dismissing Popups

- **Caveats:**

- Popups can be modal or floating:
 - If **modal**, it automatically gets activated
 - Clicking a VI in the application will cause the blocked-by-modal audible warning
 - If **floating**, “Hide when LabVIEW is not active” must be set to FALSE to get activated automatically
- Deactivation event won't fire unless VI is activated first
- Deactivation Event must be **limited to 1**





UX Considerations

Swipe / Gesture Control

- Windows gestures do not pass to LabVIEW
 - **Approximate with** Mouse Down, Mouse Move, Mouse Up, and Mouse Leave
 - Dynamically register event for **Mouse Move, Mouse Up, and Mouse Leave** on Mouse Down, Mouse Enter
- Use to replace scrolling, selection



Summary

- **Touchscreen**
 - Capacitive Screens – No Force, Can't use Gloves, Accurate
 - Resistive Screen – Uses Force, Can use Gloves, Accuracy Varies
- **UI Considerations** - Making User/Software Interactions *Accessible*
 - Make UI Elements Big Enough
 - Standardize on a Set of Controls
 - Beware the Scrollbar Sizing Gotcha
- **UX Considerations** - Making User/Software Interactions *Easier*
 - Plan to Minimize Clicks, Text Entry, Difficult Gestures
 - Balance between Ease of Use and Screen Real Estate



Summary of Links

Where do I go again?

This presentation and video will be accessible at:

- https://labviewwiki.org/wiki/Americas_CLA_Summit_2019



Summary of Links

Where do I go again?

G Community Links

- www.gcentral.org
- www.lavag.org
- www.labviewwiki.org
- www.gpackage.io
- www.gdevcon.com

Independent Source to find G Libraries (**coming soon**)

Independent Source for G Discussion

Independent G Knowledge Base

Independent G Packager/Repository

Independent Graphical Programming Conference